# CONTROL CHANNEL IMPLEMENTATION IN A PACKET SWITCHED COMMUNICATIONS NETWORK

## FIELD OF THE INVENTION

This invention relates to packet switched communications networks and more particularly to a system and method for implementing a control channel between switching nodes in such networks.

## BACKGROUND

Packet switched communications networks having multiple switching nodes connected by the Ethernet are used extensively in computer based communications systems. The Ethernet is used in local area networks (LAN), for example, to interconnect or link a plurality of clients to a client server. Ethernet implementations typically transfer data packets or frames between switching nodes over a physical medium such as a twisted copper pair, coaxial cable, fiber optic cable, etc. Early Ethernet systems operated at data rates of 10-Mb/s and typically employed coaxial cables. Second generation Ethernet systems operate at 100 Mb/s and the more recent Ethernet systems known as Gigabit Ethernet operate at data rates of 1 Gigabit per second and will typically employ fiber optic cables.

The Ethernet has been standardized by the LAN Standards Committee of the IEEE under the Ethernet standard IEEE 802.3. According to this standard the Ethernet frame has a packet format which includes a preamble, source and destination address, length of data field and the data field itself. In Ethernet the data field is a variable length which can be up to 1500 bytes or octets.

The preamble frame which is an 8 octet frame is used for synchronization in the 10 Mb/s and 100 Mb/s systems. In the Gigabit Ethernet system, however, the synchronization function is not necessary because the link is always active and the synchronization is always maintained. The present invention makes use of this preamble frame by selecting portions of the unused octet for implementing a control channel between switching nodes.

## SUMMARY OF THE INVENTION

This invention provides an implementation of a cost-free control channel between Ethernet switches by exploiting the Ethernet frame preamble. In a network of switches, it is often desirable to transfer information between devices, for example, port based flow control or priority information.

Therefore, in accordance with a first aspect of the present invention there is provided a method of implementing a control channel for exchanging information between switching devices in a packet switched communications network, comprising: selecting an unused portion of a packet format used for communicating between switching devices; and embedding control information in the unused portion.

In accordance with a second aspect of the present invention there is provided a system for implementing a control channel for use in exchanging information between switching devices in a packet switched communications network, comprising: means to select an unused portion of a packet format used to carry communication between switching devices; and means to embed control information in the unused portion.

## BRIEF DESCRIPTION OF THE DRAWINGS

The invention will now be described in greater detail with reference to the attached drawings wherein:

Figure 1 is a block diagram of a Ethernet system;

Figure 2 illustrates the Ethernet frame format;

Figure 3 illustrates a standard 64 bit preamble frame for an Ethernet frame; and

Figure 4 illustrates one possible preamble format for a Gigabit Ethernet frame with extra information embedded.

## DETAILED DESCRIPTION OF THE INVENTION

Figure 1 is a simplified version of a Ethernet system employing multiple computers connected to the Ethernet medium which may be a twisted copper pair, a coaxial cable or in a Gigabit Ethernet application an optical fiber. For reasons to be discussed in greater detail there are numerous occasions when it is desirable to transfer control information between respective computers. As stated previously, in a Gigabit Ethernet implementation the preamble frame which is typically used for synchronization purposes, is not required and it is the preamble frame which is used in the present invention to convey control information between computers or switching nodes.

Figure 2 illustrates a typical Ethernet frame format which has an eight octet preamble frame; a six octet destination address, a six octet source address; a two octet length of data field and a data field itself which may be of varying length from 0 to 1500 bytes.

The originality of the present invention arises from the simplicity, efficiency and frugality with which the control channel is implemented. By simply exploiting the

aforementioned Ethernet frame format the control channel is obtained without any resource cost. No data bandwidth is sacrificed and no functionality of the Ethernet is compromised.

As stated above the first eight bytes of an Ethernet frame comprises the preamble. Figure 3 illustrates the makeup of the eight octet frame from byte zero to byte seven. As illustrated in Figure 3 the first seven bytes of the preamble are all 10101010. The last byte of the preamble is called the start of frame delimiter (SFD) and is indicated by 10101011. As discussed above the packet header follows the preamble which includes information such as source and destination MAC addresses.

The preamble is typically intended as a synchronization pattern. It allows the receiving station to synchronize on the transmitted signals frequency. The first bytes of the preamble may be lost as a result of the phase lock latency so nothing important may be contained therein. In reality, however, this synchronization function is not necessary for a Gigabit Ethernet because in Gigabit Ethernet the link is always active and synchronization is always maintained. In other respects, however, Gigabit Ethernet is just a fast version of Ethernet. In particular, the standard Ethernet frame format is the same. For Gigabit Ethernet, the preamble of an Ethernet frame is essentially wasted bandwidth. It should be noted, however, that while the preamble frame is not required for synchronization, the entire frame is not available for holding control information. For example, a delimiter will typically be required to separate one packet from another.

According to the present invention this otherwise wasted bandwidth is reclaimed by using it to pass control information between devices connected by a Gigabit Ethernet link. Figure 4 shows one example of how this objective may be achieved.

Note that in Figure 4 the preamble has a special format by placing a special code 01010101 in byte 2. The next three bytes contain the embedded information. Also note that the high order bit of bytes 3 through 5 is set to 0 to contrast it with the standard octets 6 and 7. As note previously, a delimiter byte is required and in the implementation of Figure 4 the first three bytes are not used for control information. While, in theory, it may be acceptable to keep only the first byte as a delimiter to indicate that a new packet is coming and byte 7 as a frame delimiter, in reality, there may be errors in one or more both of these fields. To guard against this rather small probability, in this implementation bytes 1 and 2 as well as byte 6 are not used for holding control information.

Other preamble formats are possible as long as they exploit the octets that lead to wasted bandwidth in a Gigabit Ethernet. Applicants herein are not aware of any previous design which has carried packet by packet information in a control channel implemented by exploiting the Gigabit Ethernet frame preamble.

There are numerous applications where a channel for information exchange between switching devices is highly valuable.

Many of these applications fall under the category of clustering, in which two or more switches are managed as a single unit, and are expected to perform as a single unit.

For clustering to work, additional information about each packet may need to be transported among devices so that the system can behave as a unified whole. Examples of additional information which may be transported follow.

1. Flow control disable/enable: Suppose that when a packet arrives at its destination device within a switch cluster, its output port is congested. In order to ensure quality of service for the system, one possibility is to queue the packet, but send a flow control message back to the packet's original source port. On the other hand, if flow control is disabled on the packet's source port, the packet may simply be discarded. Whether a packet's original access port is enabled or disabled for flow control is one example of control information that needs to be exchanged among switches.

2. Port-based priority: In a port-based priority scheme, all packets originating from the same source port are assigned the same transmission priority. In a switch cluster, if a packet's source port and destination port are located on different devices, then the priority of a packet is lost once it crosses between devices. Under these circumstances, the only way for the transmission scheduler at the output to know the priority of the packet is via between-device communication.

3. Port trunking: In a switching system, a trunk is a group of physical ports that behaves as a single, logical port. If a packet is destined to a trunk, then a hashing algorithm is used to decide to which physical port in the trunk the packet will be forwarded. In a switch cluster, the physical ports that comprise a trunk may be located on multiple devices.

Therefore, to implement port trunking in a cluster, the hash result may need to be communicated among devices.

4. Port mirroring: The port mirroring feature in a switching system allows the received or transmitted data of any port to be copied to any other port in the system. As with port trunking, in a switch cluster, the ports that comprise a mirrored pair may be located on two different devices, again necessitating a control channel for information exchange.

5. Multicast distribution in a ring: Suppose that a cluster of switches is organized in a ring configuration, and that a multicast packet is forwarded around the ring to multiple destinations. One way for a device to know that the multicast packet has traversed the entire ring, and should now be discarded, is to compare its own ID with the packet's source device ID. If the two identifiers are equal, then the packet has made its way around the entire ring, and can be discarded. Without such a protocol, the multicast packet might loop forever around the ring. A multicast packet's source device ID is another example of control information that may need to be exchanged among switches.

This invention is not limited to Gigabit Ethernet. The same approach applies to any present or future incarnation of the Ethernet in which the frame preamble is not required for synchronization. While particular embodiments of the invention have been described and illustrated it will be apparent to one skilled in the art that numerous changes can be made without departing from the basic concept of the invention. It is to be understood that such changes will fall within the first scope of the invention as defined by the appended claims.